# Cross-language Retrieval Experiments at CLEF-2002

Aitao Chen
School of Information Management and Systems
University of California at Berkeley, CA 94720, USA
aitao@sims.berkeley.edu

**Abstract**

This paper describes monolingual, cross-language, and multilingual retrieval experiments using CLEF-2002 test collection. The paper presents a technique for incorporating blind relevance feedback into a document ranking formula based on logistic regression analysis, and a procedure for decomposing German or Dutch compounds into their component words.

## 1 Introduction

Multilingual text retrieval is the task of searching for relevant documents in a collection of documents in more than one language in response to a query, and presenting a unified ranked list of documents regardless of language. Multilingual retrieval is an extension of bilingual retrieval where the collection consists of documents in a single language that is different from the query language. Recent developments on multilingual retrieval were reported in CLEF-2000 [12], and CLEF-2001 [13]. Most of the multilingual retrieval methods fall into one of three groups. The first approach translates the source topics separately into all the document languages in the document collection. Then monolingual retrieval is carried out separately for each document language, resulting in one ranked list of documents for each document language. Finally the intermediate ranked lists of retrieved documents, one for each language, are merged to yield a combined ranked list of documents regardless of language. The second approach translates a multilingual document collection into the topic language. Then the topics are used to search against the translated document collection. The third one also translates topics to all document languages as in the first approach. The source topics and the translated topics are concatenated to form a set of multilingual topics. The multilingual topics are then searched directly against the multilingual document collection, which directly produces a ranked list of documents in all languages. The latter two approaches do not involve merging two or more ranked lists of documents, one for each document language, to form a combined ranked list of documents in all document languages. The merging task is hard and challenging. To the best of our knowledge, no effective technique has been developed yet. It appears most participating groups of the multilingual retrieval tasks in the TREC or CLEF evaluation conferences applied the first approach. Translating large collections of documents in multiple languages into topic languages requires the availability of machine translation systems that support the necessary language pairs, which is sometime problematic. For example, if the document collection consists of documents in English, French, German, Italian, and Spanish, and the topics are in English. To perform the multilingual retrieval task using English topics, one would have to translate the French, German, Italian, and Spanish documents into English. In this case, there exist translators, such as Babelfish, that can do the job. However, if the topics are in Chinese or Japanese, it may be more difficult or even not possible to find the translators to do the work. The availability of the translation resources and the need for extensive computation are factors that limit the applicability of the second approach. The third approach is appealing in that it does not require to translate the documents, and circumvents the difficult merging problem. However, there is some empirical evidence showing that the third approach is less effective than the first one [3].

We believe that three of the core components of the first approach are monolingual retrieval, topic translation, and merging. Performing multilingual retrieval requires many language resources such as stopwords, stemmers, bilingual dictionaries, machine translation systems, parallel or comparable corpora. At the same time, we see more and better language resources publicly available on the Internet. The end performance of multilingual retrieval can be affected by many factors such as monolingual retrieval performance of the document ranking algorithm, the quality and coverage of the translation resources, the availability of language-dependent stemmers and stopwords, and the effectiveness of merging algorithm. Since merging of ranked lists of documents is a challenging task, we

seek to improve multilingual retrieval performance by improving monolingual retrieval performance and exploiting translation resources publicly available on the Internet.

At CLEF 2002, we participated in the *monolingual*, *crosss-language*, and *multilingual* retrieval tasks. For monolingual task, we submitted retrieval runs for Dutch, French, German, Italian, and Spanish. For cross-language task, we submitted cross-language retrieval runs from English topics to document languages Dutch, French, German, Italian, and Spanish, one French-to-German run, and one German-to-French run. And for multilingual task, we submitted two runs using English topics. All of our runs used only the *title* and *desc* fields in the topics. The document collection for multilingual task consists of documents in English, French, German, Italian and Spanish. More details on document collections are presented below in section 5. Realizing the difficulty of merging multiple disjoint ranked lists of retrieved documents in multilingual retrieval, we have put little effort on the merging problem. We mainly worked on improving the performances of monolingual retrieval and cross-language retrieval since we believe improved performances in monolingual and cross-language retrieval should ultimately lead to better performance in multilingual retrieval. For all of our runs in cross-language and multilingual tasks, the topics was translated into document languages. The main translation resources we used are the SYSTRAN-based online machine translation system *Babelfish translation* and *L&H Power Translator Pro Version 7.0*. We also used parallel English/French texts in one of the English-to-French retrieval runs. The *Babylon* English-Dutch dictionary was used in cross-language retrieval from English to Dutch.

The same document ranking formula developed at Berkeley [4] back in 1993 was used for all retrieval runs reported in this paper. It was also used in our participation in the previous CLEF workshops. It has been shown that query expansion via blind relevance feedback can be effective in monolingual and cross-language retrieval. The Berkeley formula based on logistic regression has been used for years without blind relevance feedback. We developed a blind relevance feedback procedure for the Berkeley document ranking formula. All of our official runs were produced with blind relevance feedback. We will present a brief overview of the Berkeley document ranking formula in section 2. We will describe the blind relevance feedback procedure in section 3.

At CLEF 2001, we presented a German decompounding procedure that was hastily developed. The decompounding procedure uses a German base dictionary consisting of words that should not be further decomposed into smaller components. When a compound can be split into component words found in the base dictionary in more than one way, we choose to split up the compound so that the number of component words is the smallest. However if there two or more decompositions with the smallest number of component words, we choose the decomposition that is most likely. The probability for a decomposition of a compound is computed based on the relative frequencies of the component words in a German collection. We reported a slight decrease in German monolingual performance with German decompounding [3] at CLEF 2001. The slight decline in performance may be attributed to the fact that we kept both the original compounds and the component words resulted from decompounding in topic index. When we re-ran the same German monolingual retrieval with only the component words of compounds in the topics were retained, the average precision was improved by 8.88% with decompounding over without it [3]. Further improvements in performance brought by German decompounding were reported in [3] when a different method was used to compute the relative frequencies of component words.

At CLEF 2002, we used the improved version of the German decompounding procedure first described in [3]. A slightly different presentation of the same decompounding procedure is given in section 4. Two small changes were made in performing German retrieval with decompounding. Firstly, in both topic and document indexes, only the component words resulted from decompounding were kept. When a compound was split into component words, the compound itself was not indexed. Secondly, additional non-German words in the German base dictionary were removed. Our current base dictionary still has 762,342 words, some being non-German words and some being German compounds that should be excluded. It would take a major effort to clean up the base dictionary so that it contains only the German words that should not be further decomposed. The decompounding procedure initially developed for splitting up German compounds was also used to decompose Dutch compounds with a Dutch base dictionary.

For the submitted two official multilingual runs, one used unnormalized raw score to re-rank the documents from intermediate runs to produce the unified ranked list of documents. The other run used normalized score in the same way to produce the final list. To measure the effectiveness of different mergers, we developed an algorithm for computing the best performance that could possibly be achieved by merging multiple ranked lists of documents under the conditions that the relevances of the documents are known, and that the relative ranking of the documents in individual ranked lists is preserved in the unified ranked list. That is, if document $A$ is ranked higher than document $B$ in some ranked list, then in the unified ranked list, document $A$ should also be ranked higher that document $B$. The simple mergers based on unnormalized raw score, normalized raw score, or rank all preserve the relative ranking order. This procedure cannot be used to predict merging, however it should be useful for measuring the performance of merging algorithms. The procedure for producing optimal performance given

document relevances is presented in section 6.3.

## 2 Document Ranking

All of our retrieval runs used the same document ranking formula developed at Berkeley [4] to rank documents in response to a query. The log odds of relevance of document $D$ with respect to query $Q$ , denoted by $\log O(R|D,Q)$, is given by

$$\log O(R|D,Q) = log\frac{P(R|D,Q)}{P(\overline{R}|D,Q)} = -3.51 + 37.4 * x_1 + 0.330 * x_2 - 0.1937 * x_3 + 0.0929 * x_4$$

where $P(R|D,Q)$ is the probability that document $D$ is relevant to query $Q$, $P(\overline{R}|D,Q)$ the probability that document $D$ is irrelevant to query $Q$, which is 1.0 - $P(R|D,Q)$. The four composite variables $x_1, x_2, x_3$, and $x_4$ are defined as follows: $x_1 = \frac{1}{\sqrt{n}+1}\sum_{i=1}^{n}\frac{qtf_i}{ql+35}$, $x_2 = \frac{1}{\sqrt{n}+1}\sum_{i=1}^{n}\log\frac{dtf_i}{dl+80}$, $x_3 = \frac{1}{\sqrt{n}+1}\sum_{i=1}^{n}\log\frac{ctf_i}{cl}$, $x_4 = n$, where $n$ is the number of matching terms between a document and a query, $qtf_i$ is the within-query frequency of the $i$th matching term, $dtf_i$ is the within-document frequency of the $i$th matching term, $ctf_i$ is the occurrence frequency in a collection of the $i$th matching term, $ql$ is query length (i.e., number of terms in a query), $dl$ is document length (i.e., number of terms in a document), and $cl$ is collection length (i.e., number of terms in a test collection). If stopwords are removed from indexing, then $ql$, $dl$, and $cl$ are the query length, document length, and collection length, respectively, after removing stopwords. If the query terms are re-weighted, then $qtf_i$ is no longer the original term frequency, but the new weight, and $ql$ is the sum of the new weight values for the query terms. In the original training matrix, $qtf_i$ is the within-query term frequency, and $ql$ is the query length. Note that, unlike $x_2$ and $x_3$, the variable $x_1$ sums the "optimized" relative frequency without first taking the log over the matching terms. The relevance probability of document $D$ with respect to query $Q$ can be written as follows, given the log odds of relevance.

$$P(R|D,Q) = \frac{1}{1+e^{-logO(R|D,Q)}}$$

The documents are ranked in decreasing order by their relevance probability $P(R|D,Q)$ with respect to a query. The coefficients were determined by fitting the logistic regression model specified in $\log O(R|D,Q)$ to training data using a statistical software package. We refer readers to reference [4] for more details.

## 3 Relevance Feedback

It is well known that blind (also called pseudo) relevance feedback can substantially improve retrieval effectiveness. It is commonly implemented in research text retrieval systems. For example, see the papers of the groups who participated in the Ad Hoc tasks in TREC-7 [15] and TREC-8 [16]. Blind relevance feedback is typically performed in two stages. First, an initial search using the original queries is performed, after which a number of terms are selected from the $m$ top-ranked documents that are presumed relevant. The selected terms are merged with the original query to formulate a new query. Finally the new query is searched against the document collection to produce a final ranked list of documents. The techniques for deciding the number of terms to be selected, the number of top-ranked documents from which to extract terms, and ranking the terms varies.

The Berkeley document ranking formula has been in use for many years without blind relevance feedback. In this paper we present a technique for incorporating blind relevance feedback into the logistic regression-based document ranking framework. Some of the issues involved in implementing blind relevance feedback include determining the number of top ranked documents that will be presumed relevant and from which new terms will be extracted, ranking the selected terms and determining the number of terms that should be selected, and assigning weight to the selected terms. We refer readers to [9] for a survey of relevance feedback techniques.

Two factors are import in relevance feedback. The first one is how to select the terms from top-ranked documents after the initial search, the second is how to assign weight to the selected terms with respect to the terms in the initial query. For term selection, we assume the $m$ top-ranked documents in the initial search are relevant, and the rest of the documents in the collection are irrelevant. For the terms in the documents that are presumed relevant, we compute the odds ratio of seeing a term in the set of relevant documents and in the set of irrelevant documents. This is the term relevance weighting formula proposed by Robertson and Sparck Jones in [14]. Table 1 presents a word contingency table, where $n$ is the number of documents in the collection, $m$ the number of top-ranked

| | relevant | irrelevant | |
|---|---|---|---|
| indexed | $m_t$ | $n_t - m_t$ | $n_t$ |
| non-indexed | m - $m_t$ | n - $n_t$ - m + $m_t$ | n - $n_t$ |
| | m | n - m | n |

Table 1: A contingency table for a word.

| Initial Query | Selected Terms | Expanded Query |
|---|---|---|
| $t_1$ (1.0) | | $t_1$ (1.0) |
| $t_2$ (2.0) | $t_2$ (2*0.5) | $t_2$ (3.0) |
| $t_3$ (1.0) | $t_3$ (1*0.5) | $t_3$ (1.5) |
| | $t_4$ (0.5) | $t_4$ (0.5) |

Table 2: Query expansion.

documents after the initial search that are presumed relevant, $m_t$ the number of documents among the $m$ top-ranked documents that contain the term $t$, and $n_t$ the number of documents in the collection that contain the term $t$. Then we see from the above contingency table that the probability of finding the term $t$ in a relevant document is $\frac{m_t}{m}$, because $m_t$ documents out of the $m$ relevant documents contain the term $t$. Likewise, the probability of not finding the term $t$ in a relevant document is $\frac{m-m_t}{m}$. The odds of finding a term $t$ in a relevant document is $\frac{m_t}{m-m_t}$. Likewise, the odds of finding a term $t$ in an irrelevant document is $\frac{n_t-m_t}{n-n_t-m+m_t}$. The terms extracted from the $m$ top-ranked documents are ranked by their odds ratio which is given by

$$w_t = log\frac{m_t(n-n_t-m+m_t)}{(m-m_t)(n_t-m_t)} \quad (1)$$

For every term $t$, except for stopwords, found in the $m$ top-ranked documents, we compute its weight $w_t$ according to the above formula. Then all the terms are ranked in decreasing order by their weight $w_t$. The top-ranked $k$ terms, including the ones that are in the initial query, are added to the initial query to create a new query. For the selected top-ranked terms that are not in the initial query, the weight is set to 0.5. For those top-ranked terms that are in the initial query, the weight is set to 0.5*$t_i$, where $t_i$ is the occurrence frequency of term $t$ in the initial query. The weights are unchanged for the initial query terms that are not in the set of selected terms. The selected terms are merged with the initial query to formulate an expanded query. When a selected term is one of the query terms in the initial query, its weight in the expanded query is the sum of its weight in the initial query and its weight assigned in the term selection process. For a selected term that is not in the initial query, its weight in the final query is the same as the weight assigned in the term selection process, which is 0.5. The weights for the initial query terms that are not in the list of selected terms remain unchanged. Table 2 presents an example to illustrate how the expanded query is created from the initial query and the selected terms. The numbers in parentheses are term weights. For example, the weight for term $t_3$ in the expanded query is 3.0, since it is in the initial query with a weight value of 2.0 and it is one of the selected terms assigned the weight of 2*0.5.

Three minor changes are made to the blind relevance feedback procedure described above. First, a constant of 0.5 was added to every item in the formula used to compute the weight. Second, the selected terms must occur in at least 3 of the top-ranked $m$ documents that are presumed relevant. Third, the top-ranked two documents in the initial search remained as the top-ranked two documents in the final search. That is, the final search does not affect the ranking of the first two documents in the initial search. The rationale for not changing the top-ranked two documents is that when a query has only one or two relevant documents in the entire collection and if they are not ranked in the top in the initial search, it is unlikely these few relevant documents would be risen to the top in the second search. On the other hand, if these few relevant documents are ranked in the top in the initial search, after expansion, they are likely to be ranked lower in the final search. We believe a good strategy is to not change the ranking of the top two documents.

Note that in computing the relevance probability of a document with respect to a query in the initial search, the $ql$ is the number of terms in the initial query, and $qtf_t$ is the number of times that term $t$ occurs in the initial query. After query expansion, $qtf_t$ is no longer the raw term frequency in the initial query, instead it is now the weight of term $t$ in the expanded query, and $ql$ is the sum of the weight values of all the terms in the expanded query. For the example presented in table 2, $qtf_{t_3}$ is 1.5, and $ql$ is 6.0 (i.e., 1.0 + 3.0 + 1.5 + 0.5). The relevance clues related to documents and the collection are the same in computing relevance probability using the expanded query as in

computing relevance probability using the initial query. For all the experiments reported below, we selected the top 10 terms ranked by $w_t$ from 10 top-ranked documents in the initial search.

# 4 Decompounding

It appears most German compounds are formed by directly joining two or more words. Such examples are *Computerviren* (computer viruses), which is the concatenation of *Computer* and *Viren*, and *Sonnenenergie* (solar energy), which is formed by joining *sonnen* and *Energie* together. Sometimes a *linking element* such as *s* or *e* is inserted between two words. For example, the compound *Schönheitskönigin* (beauty queen) is derived from *Schönheit* and *königin* with *s* inserted between them. There are also cases where compounds are formed with the final letter *e* of the first word elided. For example, the compound *Erdbeben* (earthquake) is derived from *Erde* (earth) and *Beben* (trembling). When the word *Erde* is combined with the word *Atmoshpäre* to create a compound, the compound is not *Erdeatmoshpäre*, but *Erdatmoshpäre*. The final letter *e* of the word *Erde* is elided from the compound. We refer readers to, for example, [6] for discussions of German compounds formations. The example *earthquake* shows compounds are also used in English, just not nearly as commonly used as in German.

We present a German decompounding procedure in this section which will only address the cases where the compounds are directly formed by joining words and the cases where the linking element *s* is inserted. The procedure is described as follows:

1. Create a German base dictionary consisting of German words in various forms, but not compounds.
2. Decompose a German compound with respect to the base dictionary. That is, find all possible ways to break up a compound with respect to the base dictionary.
3. Choose the decomposition of the minimum number of component words.
4. If there are more than one decompositions that have the smallest number of component words, choose the one with the highest probability of decomposition. The probability of a decomposition is estimated by product of the relative frequency of the component words. More details are presented below.

For example, when the German base dictionary contains *ball, europa, fuss, fussball, meisterschaft* and others, the German compound *fussballeuropameisterschaft* can be decomposed into component words with respect to the base dictionary in two different ways as shown in Table 3. The last decomposition has the smallest number of

| | Decompositions | | | |
|---|---|---|---|---|
| 1 | fuss | ball | europa | meisterschaft |
| 2 | fussball | europa | meisterschaft | |

Table 3: Decompositions of compound *fussballeuropameisterschaft*.

component words, so the German compound *fussballeuropameisterschaft* is split into *fussball*, *europa* and *meisterschaft*. Table 4 presents another example which shows the decompositions of German compound *wintersports* with respect to a base dictionary containing *port, ports, s, sport, sports, winter, winters* and other words. The

| | Decompositions | | | log p(D) |
|---|---|---|---|---|
| 1 | winter | s | ports | -43.7002 |
| 2 | winter | sports | | -20.0786 |
| 3 | winters | ports | | -28.3584 |

Table 4: Decompositions of compound *wintersports*.

compound *wintersports* has three decompositions with respect to the base dictionary. Because two decompositions have the smallest number of component words, the rule of selecting the decomposition with the smallest number of component words cannot be applied here. We have to compute the probability of the decomposition for the decompositions with the smallest number of component words. The last column in Table 4 shows the log of the decomposition probability for all three decompositions that were computed using relative frequencies of the components words in the German test collection. According to the rule of selecting the decomposition of the highest

probability, the second decomposition should be chosen as the decomposition of the compound *wintersports*. That is, the compound *wintersports* should be split into *winter* and *sports*. Consider the decomposition of compound $c$ into $n$ component words, $c = w_1 w_2 \ldots w_n$. The probability of a decomposition is computed as follows:

$$p(c) = p(w_1)p(w_2)\ldots p(w_n) = \prod_{i=1}^{n} p(w_i)$$

where the probability of component word $w$ is computed as follows:

$$p(w_i) = \frac{tfc(w_i)}{\sum_{j=1}^{N} tfc(w_j)}$$

where $tfc(w_i)$ is the number of occurrences of word $w_i$ in a collection, $N$ is the number of unique words, including compounds, in the collection. The occurrence frequency of a word is the number of times the word occurs alone in the collection. The frequency count of a word does not include the cases where the word is a component word in a larger compound. Also, the base dictionary does not contain any words that are three-letter long or shorter except for the letter *s*. We created a German base dictionary of about 762,000 words by combining a lexicon extracted from Morphy, a German morphological analyzer [10], German wordlists found on the Internet, and German words in the CLEF-2001 German collection. In our implementation, we considered only the case where a compound is the concatenation of component words, and the case where the linking element *s* is present. Note that the number of possible decompositions of a compound is determined by what is in the base dictionary. For example, when the word *mittagessen* (lunch) is not in the base dictionary, the compound *mittagessenzeit* (lunch time) would be split into three component words *mittag* (noon), *essen* (meal), and *zeit* (time).

It is not always desirable to split up German compounds into their component words. Consider again the compound *Erdbeben*. In this case, it is probably better not to split up the compound. But in other cases like *Gemüseexporteure* (vegetable exporters), *Fußballweltmeisterschaft* (World Soccer Championship), splitting up the compounds probably is desirable since the use of the component words might retrieval additional relevant documents which are otherwise likely to be missed if only the compounds are used. In fact, we noticed that the compound *Gemüseexporteure* does not occur in the CLEF-2001 German document collection.

In general, it is conceivable that breaking up compounds is helpful. The same phrase may be spelled out in words sometimes, but as one compound other times. When a user formulate a German query, the user may not know if a phrase should appear as multi-word phrase or as one compound. An example is the German equivalent of the English phrase "European Football Cup", in the title of topic 113, the German equivalent is spelled as one compound *Fussballeuropameisterschaft*, but in the *description* field, it is *Europameisterschaft im Fußball*, yet in the *narrative* field, it is *Fußballeuropameisterschaft*. This example brings out two points in indexing German texts. First, it should be helpful to split compounds into component words. Second, normalizing the spelling of ss and ß should be helpful. Two more such examples are *Scheidungsstatistiken* and *Präsidentschaftskandidaten*. The German equivalent of "divorce statistics" is *Scheidungsstatistiken* in the *title* field of topic 115, but *Statistiken über die Scheidungsraten* in the *description* field. The German equivalent of "presidency candidates" is *Präsidentschaftskandidaten* in *title* field of topic 135, but *Kandidat für das Präsidentenamt* in the *description* field of the same topic. The German equivalent for "Nobel price winner for literature" is *Literaturnobelpreisträger*, in the "Der Spiegel" German collection, we find variants of *Literatur-Nobelpreisträger*, *Literaturnobelpreis-Trgerin*. Literaturnobelpreis sometimes appears as "Nobelpreis für Literatur".

# 5 Test Collection

The document collection for the multilingual IR task consists of documents in five languages: English, French, German, Italian, and Spanish. The collection has about 750,000 documents which are newspaper articles published in 1994 except that part of the *Der Spiegel* was published in 1995. The distribution of documents among the five document languages is presented in Table 5. A set of 50 topics was developed and released in more than 10 languages, including Dutch, English, French, German, Italian, and Spanish. A topic has three parts: 1) *title*, a short description of information need; 2) *description*, a sentence-long description of information need; and 3) *narrative*, specifying document relevance criteria. More details about the test collection are presented in [13]. The multilingual IR task at CLEF 2002 was concerned with searching the collection consisting of English, French, German, Italian, and Spanish documents for relevant documents, and returning a combined, ranked list of documents in any document language in response to a query.

| Language | Name | No. of documents | Size (MB) |
|---|---|---|---|
| English | Los Angeles Times | 113,005 | 425 |
| French | Le Monde | 44,013 | 157 |
| | SDA French | 43,178 | 86 |
| German | Frankfurter Rundschau | 139,715 | 320 |
| | Der Spiegel | 13,979 | 63 |
| | SDA German | 71,677 | 144 |
| Italian | La Stampa | 58,051 | 193 |
| | SDA Italian | 50,527 | 85 |
| Spanish | EFE | 215,738 | 509 |
| Dutch | RC Handelsblad | 84,121 | 299 |
| | Algemeen Dagblad | 106,483 | 241 |

Table 5: Part of the CLEF 2002 document sets.

# 6 Experimental Results

All retrieval runs reported in this paper used only the *title* and *description* fields in the topics. The ids and average precision values of the official runs are presented in bold face, other runs are unofficial ones.

## 6.1 Monolingual retrieval experiments

In this section we present the results of monolingual retrieval. We created a stopwords list for each document language. In indexing, the stopwords were removed from both documents and topics. Additional words such as *relevant* and *document* were removed from topics. The words in all six languages were stemmed using Muscat stemmers downloaded from http://open.muscat.com. For automatic query expansion, the top-ranked 10 terms from the top-ranked 10 documents after the initial search were combined with the original query to create the expanded query. For Dutch and German monolingual runs, the compounds were split into their component words, and only their component words were retained in document and topic indexing. All the monolingual runs included automatic query expansion via the relevance feedback procedure described in section 3. Table 6 presents the monolingual retrieval results for six document languages. The last column labeled *change* shows the improvement of average precision with blind relevance feedback over without it. As table 6 shows, query expansion increased the average precision of the monolingual runs for all six languages, the improvement ranging from 6.42% for Spanish to 19.42% for French. There are no relevant Italian documents for topic 120, and no relevant English documents for

| | | without expansion | | with expansion | | |
|---|---|---|---|---|---|---|
| run id | language | recall | precision | recall | precision | change |
| bky2moen | English | 765/821 | 0.5084 | 793/821 | 0.5602 | 10.19% |
| **bky2monl** | Dutch | 1633/1862 | 0.4446 | 1734/1862 | **0.4847** | 9.02% |
| **bky2mofr** | French | 1277/1383 | 0.4347 | 1354/1383 | **0.5191** | 19.42% |
| **bky2mode** | German | 1696/1938 | 0.4393 | 1807/1938 | **0.5234** | 19.14% |
| **bky2moit** | Italian | 994/1072 | 0.4169 | 1024/1072 | **0.4750** | 13.94% |
| **bky2moes** | Spanish | 2531/2854 | 0.5016 | 2673/2854 | **0.5338** | 6.42% |

Table 6: Monolingual IR performance.

topics 93, 96, 101, 110, 117, 118, 127 and 132.

For the German monolingual runs, compounds were decomposed into their component words by applying the decompounding procedure described above. Only component words of the decomposed compounds were kept in document and topic indexing. Table 7 presents the performance of German monolingual retrieval with three different features which are decompounding, stemming, and query expansion. The features are implemented in the order of *decompounding*, *stemming*, and *query expansion*. For example, when *decompounding* and *stemming* are present, the compounds are split into component words first, then the components are stemmed. The table shows when any one of the three features is present, the average precision improves from 4.94% to 19.73% over

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| features | none | decomp | stem | expan | decomp+stem | decomp+expan | stem+expan | decomp+stem+expan |
| avg prec | 0.3462 | 0.3859 | 0.3633 | 0.4145 | 0.4393 | 0.4517 | 0.4393 | **0.5234** |
| recall | 1359 | 1577 | 1500 | 1575 | 1696 | 1752 | 1702 | 1807 |
| change | baseline | +11.47% | +4.94% | +19.73% | +26.89% | +30.47% | +26.89% | +51.18% |

Table 7: German monolingual retrieval performance. The total number of German relevant documents for 50 topics is 1938.

the baseline performance when none of the features is present. When two of the three features are included in retrieval, the improvement in precision ranges from 26.89% to 30.47%. And when all three features are present, the average precision is 51.18% better than the baseline performance. It is interesting to see the three features are complementary. That is, the improvement brought by each individual feature is not diminished by the presence of the other two features. Without decompounding, stemming alone improved the average precision by 4.94%. However with decompounding, stemming improved the average precision from 0.3859 to 0.4393, an increase of 13.84%. Stemming became more effective because of decompounding. Decompounding alone improved the average precision by 11.47% for German monolingual retrieval.

| | compounds | component | words | | compounds | component | words |
|---|---|---|---|---|---|---|---|
| 1 | absatzkrise | absatz | krise | 2 | atemwege | atem | wege |
| 3 | autoindustrie | auto | industrie | 4 | automobilindustrie | automobil | industrie |
| 5 | bandleaders | band | leaders | 6 | bronchialasthma | bronchial | asthma |
| 7 | bürgerkrieg | bürger | krieg | 8 | computeranimation | computer | animation |
| 9 | computeranimationen | computer | animationen | 10 | computersicherheit | computer | sicherheit |
| 11 | durchbrüche | durch | brüche | 12 | eigentumsrechte | eigentums | rechte |
| 13 | eurofighter | euro | fighter | 14 | europameisterschaft | europa | meisterschaft |
| 15 | filmfestspielen | film | festspielen | 16 | filmindustrie | film | industrie |
| 17 | fischereiquoten | fischerei | quoten | 18 | fremdsprachigen | fremd | sprachigen |
| 19 | fremdwörter | fremd | wörter | 20 | goldmedaille | gold | medaille |
| 21 | handynutzung | handy | nutzung | 22 | interessenkonflikt | interessen | konflikt |
| 23 | interessenkonflikts | interessen | konflikts | 24 | menschenrechte | menschen | rechte |
| 25 | mobiltelefone | mobil | telefone | 26 | nahrungskette | nahrungs | kette |
| 27 | netzwerken | netz | werken | 28 | nordamerika | nord | amerika |
| 29 | nordamerikanische | nord | amerikanische | 30 | pelzindustrie | pelz | industrie |
| 31 | präsidentschaftskandidaten | präsidentschafts | kandidaten | 32 | premierministers | premier | ministers |
| 33 | scheidungsraten | scheidungs | raten | 34 | scheidungsstatistiken | scheidungs | statistiken |
| 35 | sicherheitspolitik | sicherheits | politik | 36 | spionagefall | spionage | fall |
| 37 | spionagefalles | spionage | falles | 38 | sternensystemen | sternen | systemen |
| 39 | verkaufszahlen | verkaufs | zahlen | 40 | volksbefragung | volks | befragung |
| 41 | winterspielen | winter | spielen | 42 | wintersports | winter | sports |
| 43 | wirtschaftsembargos | wirtschafts | embargos | 44 | wirtschaftspolitik | wirtschafts | politik |
| 45 | zeitplan | zeit | plan | 46 | zurücktreten | zurück | treten |
| 47 | einheitswährung | einheit | s | währung | | | |
| 48 | fischfangquoten | fisch | fang | quoten | | | |
| 49 | fussballeuropameisterschaft | fussball | europa | meisterschaft | | | |
| 50 | geographos | geog | rapho | s | | | |
| 51 | literaturnobelpreisträgers | literatur | nobel | preisträgers | | | |
| 52 | schönheitswettbewerbe | schönheit | s | wettbewerbe | | | |
| 53 | schönheitswettbewerben | schönheit | s | wettbewerben | | | |

Table 8: German words in *title* or *desc* fields of the topics that are split into component words.

Table 8 presents the German words in the *title* or *desc* fields of the topics that were split into component words using the decompounding procedure described in section 4. The column labeled *component words* shows the component words of the decomposed compounds. The German word *eurofighter* was split into *euro* and *fighter* since both component words are in the base dictionary, and the word *eurofighter* is not. Including the word *eurofigher* in the base dictionary will prevent it from being split into component words. The word *geographos* was decomposed into *geog*, *rapho*, and *s* for the same reason that the component words are in the base dictionary. Two topic words, *lateinamerika* and *zivilbevölkerung*, were not split into component words because both are present in our base dictionary which is far from being perfect. For the same reason, the *preisträgers* was not decomposed into *preis* and *trägers*. An ideal base dictionary should contain all and only the words that should not be further split into smaller component words. Our current decompounding procedure does not split words in the base dictionary

into smaller component words. When the two compounds, *lateinamerika* and *zivilbevölkerung*, are removed from the base dictionary, *lateinamerika* is split into *latein* and *amerika*, and *zivilbevölkerung* into *zivil* and *bevölkerung*. The topic word *südjemen* was not split into *süd* and *jemen* because our base dictionary does not contain words that are three-letter long or shorter. The majority of the errors in decompounding are caused by the incompleteness of the base dictionary or the presence of compound words in the base dictionary.

We used a Dutch stopword list of 1326 words downloaded from http://clef.iei.pi.cnr.it:2002/ for Dutch monolingual retrieval. After removing stopwords, the Dutch words were stemmed using the muscat Dutch stemmer. For Dutch decompounding, we used a Dutch wordlist of 223,557 words [1]. From this wordlist we created a Dutch base dictionary of 210,639 by manually breaking up the long words that appear to be compounds. It appears that many Dutch compound words remain in the base dictionary. Like the German base dictionary, an ideal Dutch base dictionary should include all and only the words that should not be further decomposed into smaller component words. The Dutch words in the *topics* or *desc* fields of the topics were split into component words using the same procedure as for German decompounding. Like German decompounding, the words in the Dutch base dictionary are not decomposed. The source wordlist files contain a list of country names, which should have beed added to the

| | compounds | component | words | | compounds | component | words |
|---|---|---|---|---|---|---|---|
| 1 | rijkspolitie | rijks | politie | 2 | belangenverstrengeling | belangen | verstrengeling |
| 3 | sterrenstelsels | sterren | stelsels | 4 | bontsector | bont | sector |
| 5 | nobelprijs | nobel | prijs | 6 | verkoopaantallen | verkoop | aantallen |
| 7 | grungerock | grunge | rock | 8 | spionagezaak | spionage | zaak |
| 9 | frankrijk | frank | rijk | 10 | echtscheidingscijfers | echtscheidings | cijfers |
| 11 | oproepkaart | oproep | kaart | 12 | autofabrikanten | auto | fabrikanten |
| 13 | handelsembargo | handels | embargo | 14 | internationale | inter | nationale |
| 15 | duitsland | duit | s land | 16 | computerbeveiliging | computer | beveiliging |
| 17 | filmindustrie | film | industrie | 18 | veiligheidsbeleid | veiligheids | beleid |
| 19 | netwerktoegang | veiligheids | beleid | 20 | filmfestival | film | festival |
| 21 | omzetcrisis | omzet | crisis | 22 | computeranimatie | computer | animatie |
| 23 | tijdschema | tijd | schema | | | | |

Table 9: Dutch words in *title* or *desc* fields of the topics that are split into component words.

Dutch base dictionary. The Dutch words *frankrijk* and *duitsland* were split into component words because they are not in the base dictionary. For the same reason, the word *internationale* was decomposed. It appears compound words in Dutch are not as common as in German. Like in German indexing, when a compound was split into component words, only the component words were retained in the index. Table 10 presents the performance of

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| features | none | decomp | stem | expan | decomp+stem | decomp+expan | stem+expan | decomp+stem+expan |
| avg prec | 0.4021 | 0.4186 | 0.4281 | 0.4669 | 0.4446 | 0.4721 | 0.4770 | **0.4847** |
| recall | 1562 | 1623 | 1584 | 1614 | 1633 | 1727 | 1702 | 1734 |
| change | baseline | +4.10% | +6.47% | +16.12% | +10.57% | +17.41% | +18.63 | +20.54% |

Table 10: Dutch monolingual retrieval performance on CLEF-2002 test set. The total number of Dutch relevant documents for the 50 topics of CLEF 2002 is 1862.

Dutch monolingual retrieval under various conditions. With no stemming and expansion, Dutch decompounding improved the average precision by 4.10%. Together the three features improved the average precision by 20.54% over the base performance when none of the features is implemented.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| features | none | decomp | stem | expan | decomp+stem | decomp+expan | stem+expan | decomp+stem+expan |
| avg prec | 0.3239 | 0.3676 | 0.3587 | 0.3471 | 0.4165 | 0.3822 | 0.3887 | 0.4372 |
| change | baseline | +13.49% | +10.74% | +7.16% | +28.59% | +18.00% | +20.01% | +34.98% |

Table 11: Dutch monolingual retrieval performance on CLEF-2001 test set. The total number of Dutch relevant documents for the 50 topics of CLEF 2001 is 1224.

For comparison, table 11 presents the Dutch monolingual performance on the CLEF 2001 test set. Decompounding alone improved the average precision by 13.49%. Topic 88 of CELF 2001 is about *mad cow diseases in*

[1] downloaded from ftp://archive.cs.ruu.nl/pub/UNIX/ispell/words.dutch.gz

*Europe*. The Dutch equivalent of *mad cow diseases* is *gekkekoeienziekte* in the topic, but never occurs in the Dutch collection. Without decompounding, the precision for this topic is 0.1625, and with decompounding, the precision increased to 0.3216. The precision for topic 90 which *vegetable exporters* is 0.0128 without decompounding. This topic contains two compound words, *Groentenexporteurs* and *diepvriesgroenten*. The former one which is perhaps the most important term for this topic never occurs in the Dutch document collection. After decompounding, the precision for this topic increased to 0.3443. Topic 55 contains two important compound words, *Alpenverkeersplan* and *Alpeninitiatief*. Both never occur in the Dutch document collection. The precision for this topic is 0.0746 without decompounding, and increased to 0.2137 after decompounding.

## 6.2 Cross-language Retrieval Experiments

A major factor affecting the end performance of cross-language retrieval and multilingual retrieval is the quality of translation resources. In this section, we evaluate the effectiveness of three different translation resources: automatic machine translation systems, parallel corpora, and bilingual dictionaries. Two of the issues in translating topics are 1) determining the number of translations to retain when multiple candidate translations are available; and 2) assigning weights to the selected translations [8]. When machine translation systems are used to translate topics, these two issues are resolved automatically by the machine translation systems, since they provides only one translation for each word. However, when bilingual dictionaries or parallel corpora are used to translate topics, often for a source word, there may be several alternative translations.

### 6.2.1 CLIR Using MT

In this section, we evaluate two machine translation systems, online Babelfish translation [2] and L&H Power Translator Pro, version 7.0, for translating topics in CLIR. We used both translators to translate the 50 English topics into French, Italian, German, and Spanish. For each language, both sets of translations were preprocessed in the same way. Table 12 presents the CLIR retrieval performances for all the official runs and additional runs. The

| | | | | without expansion | with expansion | |
|---|---|---|---|---|---|---|
| run id | topic | document | resources | precision | precision | change |
| **bky2bienfr** | English | French | Babelfish + L&H | 0.4118 | **0.4773** | +15.91% |
| **bky2bienfr2** | English | French | Systran + L&H + Parallel Texts | 0.4223 | **0.4744** | +12.34% |
| bky2bienfr3 | English | French | Babelfish | 0.3731 | 0.4583 | +22.84% |
| bky2bienfr4 | English | French | L&H | 0.3951 | 0.4652 | +17.74% |
| bky2bienfr5 | English | French | Parallel texts | 0.3835 | 0.4529 | +18.10% |
| **bky2bidefr** | German | French | Babelfish | 0.3437 | **0.4124** | +19.99% |
| **bky2biende** | English | German | Babelfish + L&H | 0.3561 | **0.4479** | +25.78% |
| bky2biende1 | English | German | Babelfish | 0.3229 | 0.4091 | +26.70% |
| bky2biende2 | English | German | L&H | 0.3555 | 0.4449 | +25.15% |
| **bky2bifrde** | French | German | Babelfish | 0.3679 | **0.4759** | +29.36% |
| **bky2bienit** | English | Italian | Babelfish + L&H | 0.3608 | **0.4090** | +13.36% |
| bky2bienit1 | English | Italian | Babelfish | 0.3239 | 0.3634 | +12.20% |
| bky2bienit2 | English | Italian | L&H | 0.3412 | 0.3974 | +16.47% |
| **bky2bienes** | English | Spanish | Babelfish + L&H | 0.4090 | **0.4567** | +11.66% |
| bky2bienes1 | English | Spanish | Babelfish | 0.3649 | 0.4108 | +12.58% |
| bky2bienes2 | English | Spanish | L&H | 0.4111 | 0.4557 | +10.85% |
| **bky2biennl** | English | Dutch | Babylon | 0.2564 | **0.3199** | +24.77% |

Table 12: Performance of cross-language retrieval runs. The ids and average precision values for the official runs are in bold face.

ids and average precision values for the official runs are in bold face. Last column in table 12 shows the improvement of average precision with query expansion over without it. When both L&H Translator and Babelfish

[2] publicly available at http://babelfish.altavista.com/

were used in cross-language retrieval from English to French, German, Italian and Spanish, the translation from L&H Translator and the translation from Babelfish were combined by topic. The term frequencies in the combined topics were reduced by half so that the combined topics were comparable in length to the source English topics. Then the combined translations were used to search the document collection for relevant documents as in monolingual retrieval. For example, for the English-to-Italian run *bky2bienit*, we first translated the source English topics into Italian using L&H Translator and Babelfish. The Italian translations produced by L&H Translator and the Italian translations produced by Babelfish were combined by topic. Then the combined, translated Italian topics with term frequencies reduced by half were used to search the Italian document collections. The *bky2bienfr*, *bky2biende*, *bky2bienes* CLIR runs from English were all produced in the same way as the *bky2bienit* run. For English or French to German cross-language retrieval runs, the words in *title* or *desc* fields of the translated German topics were decompounded. For all cross-language runs, words were stemmed after removing stopwords like in monolingual retrieval. The English-to-French run *bky2bienfr2* was produced by merging the *bky2bienfr* run and the *bky2bienfr5* run which used parallel corpora as the sole translation resource. More discussion about the use of parallel corpora will be presented below.

All the cross-language runs applied blind relevance feedback. The top-ranked 10 terms from the top-ranked 10 documents after the initial search were combined with the initial query to formulate an expanded query. The results presented in table 12 show that query expansion improved the average precision for the official runs from 10.85% to 29.36%. The L&H Translator performed better than Babelfish for cross-language retrieval from English to French, German, Italian and Spanish. Combining the translations from L&H Translator and Babalfish performed slightly better than using only the translations from L&H translator.

We notices a number of error in translating English to Italian using Babelfish. For example, the English text *Super G* which was translated into *Superg*, *U.N.* and *U.S.-Russian* were not translated. While the phrase *Southern Yemen* in the *desc* field was correctly translated into *Südyemen*, the same phrase in the *title* field became *SüdcYemen*. Decompounding is helpful in monolingual retrieval, it is also helpful in cross-language retrieval to German from

| | | | no decompounding | decompounding | |
|---|---|---|---|---|---|
| source | target | translator | average precision | average precision | change |
| English | German | L&H Translator | 0.2776 | 0.3009 | 8.4% |
| English | German | Babelfish | 0.2554 | 0.2906 | 13.78% |
| French | German | Babelfish | 0.2774 | 0.3092 | 11.46% |

Table 13: Effectiveness of decompounding in cross-language retrieval to German. All runs were performed without stemming and query expansion.

other languages such as English. An English phrase of two words may be translated into a German phrase of two words, or into a compound. For examples, in topic 111, the English phrase *computer animation* in *title* became *ComputerAnimation*, and *Computer Animation* in *desc*. In topic 109, the English phrase *Computer Security* became *Computer-Sicherheit* in the title, but the same phrase in lower case in *desc* became *Computersicherheit*. Table 13 shows the performances of three cross-language retrieval to German with and without decompounding. The improvement in average precision ranges from 8.4% to 13.78%.

### 6.2.2 English-French CLIR Using Parallel Corpora

We created a French-English bilingual lexicon from the Canadian Hansard (the recordings of the debates of the House for the period of 1994 to 2001). The texts are in English and French. We first aligned the Hansard corpus at the sentence level using the length-based algorithm proposed by Gale and Church [7], resulting in about two million aligned French/English sentence pairs. To speed up the training (i.e, estimating word translation probabilities), we extracted and used only the sentence pairs that contain at least one English topic word in CLEF-2001 topics. A number of preprocessing steps were carried out prior to the training. First, we removed the English stopwords from the English sentences, and French stopwords from the French sentences. Secondly, we changed the variants of a word into its base form. For English, we used a morphological analyzer described in [5]. For French, we used a French morphological analyzer named DICO. Each of the packages contains a list of words together with their morphological analyses. Thirdly, we discarded the sentence pairs in which one of the sentence has 40 or more words after removing stopwords, and the sentence pairs in which the length ratio of the English sentence over the French sentence is below .7 or above 1.5. The average length ratio of English text over French text is approximately 1.0. Since sentence alignment is not perfect, some mis-alignments are unavoidable. Hence there may be sentence

pairs in which the length ratios that deviate far from the average length ratio. After the preprocessing, only 706,210 pairs of aligned sentences remained. The remaining aligned sentence pairs were fed to GIZA++ for estimating English-to-French word translation probabilities. GIZA++ toolkit is an extension to the EGYPT toolkit [1] which was based on the statistical machine translation models described in [2]. Readers are referred to [11] for more details on GIZA++. The whole training phase took about 24 hours on a Sun Microsystem Sparc server machine. Table 14 shows the first three French translations produced by GIZA++ for some of the words in the English topics.

| English | French translations | Translation probability | English | French translation | Translation probability |
|---|---|---|---|---|---|
| amnesty | amnistier | 0.960881 | independence | indpendance | 0.762385 |
| | amnistie | 0.032554 | | autonomie | 0.142249 |
| | amnesty | 0.006565 | | indpendant | 0.032079 |
| asthma | asthme | 0.902453 | lead | mener | 0.128457 |
| | asthma | 0.053307 | | conduire | 0.076652 |
| | atteindre | 0.044240 | | amener | 0.066278 |
| car | voiturer | 0.251941 | phone | téléphoner | 0.419111 |
| | automobile | 0.214175 | | téléphonique | 0.194628 |
| | voiture | 0.160644 | | appeler | 0.123656 |
| computer | informatique | 0.438414 | prime | ministre | 0.898780 |
| | ordinateur | 0.434168 | | chrtien | 0.049399 |
| | informatiser | 0.021902 | | principal | 0.003372 |
| conflict | conflit | 0.873595 | race | race | 0.598102 |
| | guerre | 0.016048 | | courser | 0.182982 |
| | contradictoire | 0.012773 | | racial | 0.053363 |
| currency | monnayer | 0.455730 | rock | rock | 0.889616 |
| | deviser | 0.108036 | | rocher | 0.015060 |
| | devise | 0.106799 | | pierre | 0.010083 |
| fall | automne | 0.323739 | right | droit | 0.973834 |
| | tomber | 0.081521 | | rights | 0.002897 |
| | relever | 0.064848 | | charte | 0.001411 |
| film | film | 0.722327 | sanction | sanction | 0.600641 |
| | cinmatographique | 0.105770 | | sanctionner | 0.147880 |
| | cinma | 0.058341 | | approuver | 0.076667 |
| economic | économique | 0.830063 | star | star | 0.624963 |
| | économie | 0.104932 | | toile | 0.130342 |
| | financier | 0.010520 | | toiler | 0.077801 |

Table 14: English to French word translation probabilities estimated from parallel corpora using a statistical machine translation toolkit.

The French translations are ranked in descending order by the probability of translating from an English word into French words. In translating an English word into French, we selected only one French word, the one of the highest translation probability, as the translation. The English topics were translated into French word-by-word, then the translated French topics were used in producing the English-to-French run labeld *bky2bienfr5* in table 12. Without query expansion, the parallel corpus-based English-French CLIR performance was slightly better than that of using Babelfish, but slightly lower than that of using L&H translator.

The CLEF 2002 English topics contain a number of polysemous words such as *cup*, *fall*, *interest*, *lead*, *race*, *right*, *rock*, *star*, and the like. The word *fall* in the context of *fall in sale of cars* in topic 106 has the meaning of declining. However, the most likely French translation for *fall* as table 14 shows is *automne*, meaning *autumn* in English. The word *race* in *ski race* in topic 102 or in *race car* in topic 121 has the meaning of contest or competition in speed. Again the French word of the highest translation probability is *race*, meaning human race in English. The corrent French translation for the sense of *race* in *ski race* or *car race* should be *course*. The word *star* in topic 129 means a plant or celestial body, while in topic 123 in the context of *pop star*, it means a famous performer. The correct translation for *star* in topic 129 should be *étoile*, instead of the most likely translation *star*, which is the correct French word for the sense of *star* in *pop star*. The word *rock* in topic 130 has the same sense as *rock* in *rock music*, not the sense of *stone*. The correct translation for *rock* in topic 130 should be *rocke*. In the same topic, the word *lead* in *lead singer* means someone in the leading role, not the metal. These examples show that taking the French word of the highest translation probability as the translation for an English word is overly simplified. Choosing the right French translations would require word sense disambiguation.

### 6.2.3 CLIR using bilingual dictionary

For the only English-to-Dutch run *bky2biennl*, the English topics were translated into Dutch by looking up each English topic word, excluding stopwords, in the online English-Dutch dictionary *Babylon* [3]. All the Dutch words in the dictionary lookup results were retained except for Dutch stopwords. The Dutch compound words were split into component words. If translating an English topic word resulted in $m$ Dutch words, then all translated Dutch words of the English word received the same weight $\frac{1}{m}$, i.e., the translated Dutch words were weighted uniformly. The average precision of the English-to-Dutch run is 0.3199, which is much lower than 0.4847 for Dutch monolingual retrieval.

## 6.3 Multilingual Retrieval Experiments

In this section, we describe our multilingual retrieval experiments using the English topics (only *title* and *description* fields were indexed). As mentioned in the cross-language experiments section above, we translated the English topics into the other four document languages which are French, German, Italian, and Spanish using Babelfish and L&H Translator. A separate index was created for each of the five document languages. For the multilingual retrieval runs, we merged five ranked lists of documents, one resulted from English monolingual retrieval and four resulted from cross-language retrieval from English to the other four document languages, to produce a unified ranked list of documents regardless of language.

A fundamental difference between merging in monolingual retrieval or cross-language retrieval and merging in multilingual retrieval is that in monolingual or cross-language retrieval, documents for individual ranked lists are from the same collection, while in multilingual retrieval, the documents for individual ranked lists are from different collections. For monolingual or cross-language retrieval, if we assume that documents appearing on more than one ranked list are more likely to be relevant than the ones appearing on a single ranked list, then we should rank the documents appearing on multiple ranked lists in higher position in the merged ranked list of documents. A simple way to accomplish this is to sum the probability of relevance for the documents appearing on multiple ranked lists while the probabilities of relevance for the documents appearing on a single list remain the same. After summing up the probabilities, the documents are re-ranked in descending order by combined probability of relevance. In multilingual retrieval merging, since the documents on the individual ranked lists are all different, we cannot use multiple appearances of a document in the ranked lists as evidence to promote its rank in the final ranked list. The problem of merging multiple ranked lists of documents in multilingual retrieval is closely linked to estimating probability of relevance. If the estimates of probability of relevance are accurate and well calibrated, then one can simply combine the individual ranked lists and then re-rank the combined list by the raw probability of relevance. In practice, estimating relevance probabilities is a hard problem.

We looked at the estimated probabilities of relevance produced using the ranking formula described in section 2 for the CLEF 2001 topics to see if there is a linear relationship between the number of relevant documents and the number of documents whose estimated probabilities of relevance are above some threshold. Figure 1 shows the scatter plot of the number of retrieved documents whose estimated relevance probabilities are above 0.37 versus the number relevant documents for the same topic. Each dot in the figure represents one French topic. The ranked list of documents was produced using the 50 French topics of CLEF 2001 to search against the French collection with query expansion. The top-ranked 10 terms from top-ranked 10 documents in the initial search were merged with initial query to create the expanded query. The threshold of 0.37 was chosen so that the total number of documents for all 50 topics whose estimated relevance probabilities are above the threshold is close to the total number of relevant documents for the same set of topics. If the estimated probabilities are good, the dots in the figure would appear along the diagonal line. The figure shows there is no linear relationship between the number of retrieved documents whose relevance probabilities are above the threshold and the number of relevant documents for the same topic. This implies one cannot use the raw relevance probabilities to directly estimate the number of relevant documents for a topic in a test document collection.

There are a few simple ways to merge ranked lists of documents from different collections. Here we will evaluate two of them. The first method is to combine all ranked lists, sort the combined list by the raw relevance score, then take the top 1000 documents per topic. The second method is to normalize the relevance score for each topic, dividing all relevance scores by the relevance score of the top most ranked document for the same topic. Table 15 presents the multilingual retrieval performance with different merging strategies. The multilingual runs were produced by merging from five runs: bky2moen (English-English, 0.5602), bky2bienfr (English-French, 0.4773), bky2biende (English-German, 0.4479), bky2bienit (English-Italian, 0.4090), and bky2bienes (English-Spanish, 0.4567). The run *bky2muen1* was produced by ranking the documents by the unnormalized relevance

[3] available at http://www.babylon.com

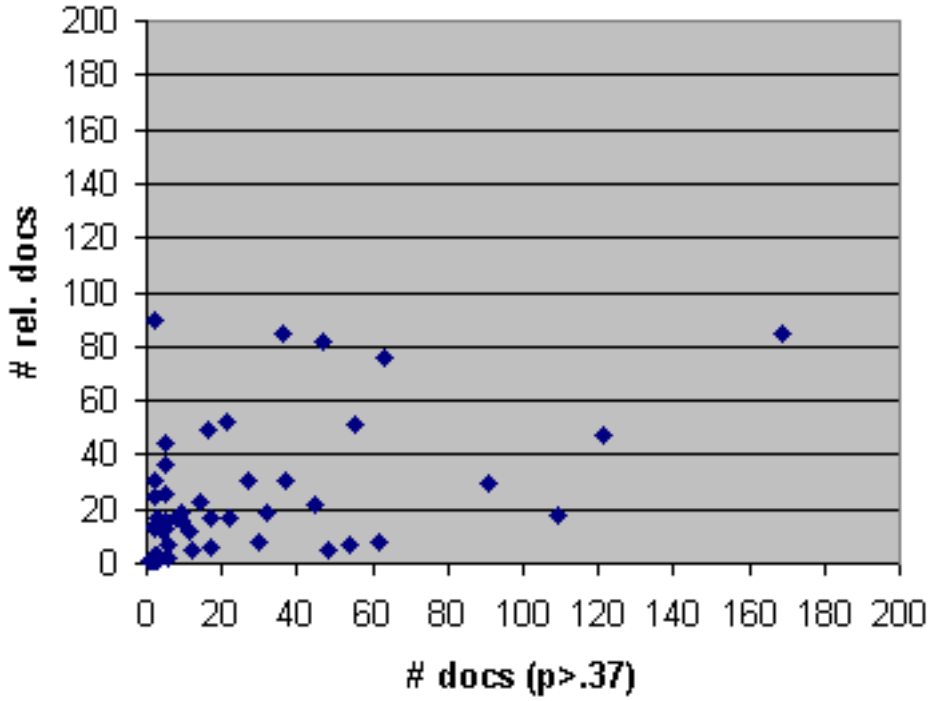


Figure 1: Number of retrieved documents with relevance probability over .37 versus the number of relevant documents for the same topic.

| run id | topic language | topic fields | merging strategy | recall | precision |
|---|---|---|---|---|---|
| **bky2muen1** | English | title,desc | Direct merging | 5880/8068 | **0.3762** |
| **bky2muen2** | English | title,desc | Normalized merging | 5765/8068 | **0.3570** |

Table 15: Multilingual retrieval performance for different merging strategies. The five runs from which the the multilingual runs were produced are *bky2moen*, *bky2bienfr*, *bky2biende*,*bky2bienit*,*bky2bienes*.

probabilities after combining the individual runs. And the run *bky2muen2* was produced in the same way except that the relevance probabilities were normalized before merging. For each topic, the relevance probabilities of the documents was divided by the relevance probability of the highest-ranked document for the same topic. The simplest direct merging outperformed the score normalizing merging strategy. We did two things to make the relevance probabilities of documents from different language collections comparable to each other. Firstly, as mentioned in section 6.2.3, after concatenating the topic translations from two translators, we reduced the term frequencies by half so that the translated topics are close to the source English topics in length. Secondly, in query expansion, we took the same number of terms (i.e, 10) from the same number of top-ranked documents (i.e, 10) after the initial search for all five individual runs that were used to produce the multilingual runs.

In the remainder of this section, we present a procedure for computing the optimal performance that could possibly be achieved under the constraint that the relative ranking of the documents in the individual ranked lists is preserved. This procedure assumes that the relevances of documents are known, thus it is not useful to to predict ranks of documents in the final ranked list for multilingual retrieval. However, knowing the upper-bound performance for a set of ranked lists of documents and the related document relevances is useful in measuring the performance of different merging strategies. We will use an example to explain the procedure. Let us assume we are going to merge three runs labeled $A$, $B$ and $C$, as shown in table 16. The relevant documents are marked with an '*'. We want to find a combined ranked list such that the average precision is maximized without changing the relative rank order of the documents on the same ranked list. First we transform the individual runs shown in table 16 into the form shown in table 17 by grouping the consecutive irrelevant and relevant documents. Each entry in table 17 has the form $(m, n)\{d_i, d_{i+1}, \ldots, d_j\}$, where $d_i$ is the id of the document ranked in the $i$th position in the original ranking. $\{d_i, d_{i+1}, \ldots, d_j\}$ denotes a set of consecutive irrelevant and relevant documents

| | Run A | Run B | Run C |
|---|---|---|---|
| 1 | $A_1$* | $B_1$ | $C_1$ |
| 2 | $A_2$ | $B_2$ | $C_2$* |
| 3 | $A_3$* | $B_3$* | $C_3$* |
| 4 | $A_4$ | $B_4$ | $C_4$* |

Table 16: Individual ranks. Relevant documents are marked with *

ranked in position from $i$ to $j$, inclusive. $m$ is the number of irrelevant documents in the set, and $n$ is the number of relevant documents in the set. For example, the entry (2,1) $\{B_1,B_2,B_3\}$ means the set $\{B_1,B_2,B_3\}$ has two irrelevant documents, $B_1$ and $B_2$, and one relevant document, $B_3$. After the transformation, the procedure can be

| set | Run A | Run B | Run C |
|---|---|---|---|
| 1 | (0,1) $\{A_1\}$ | (2,1) $\{B_1,B_2,B_3\}$ | (1,3) $\{C_1,C_2,C_3,C_4\}$ |
| 2 | (1,1) $\{A_2,A_3\}$ | (1,0) $\{B_4\}$ | |
| 3 | (1,0) $\{A_3\}$ | | |

Table 17: Ranked lists after transformation.

implemented in four steps.

Step 1: Let the *active* set consist of the first set in the individual lists that contains at least one relevant document. For the example presented in table 17, the initial *active* set is $\{$(0,1) $\{A_1\}$, (2,1) $\{B_1,B_2,B_3\}$, (1,3) $\{C_1,C_2,C_3,C_4\}\}$

Step 2: Choose the element in the *active* set with the smallest number of irrelevant documents. If there are two or more elements with the smallest number of irrelevant documents, then choose the element that also contains the largest number of relevant documents. If there are two or more elements with the same smallest number of irrelevant documents and the same largest number of relevant documents in the current *active* set, randomly choose one of them. Append the selected element to the final ranked list. If the next set appearing immediately after the selected element contains at least one relevant document, then add the next set to the current *active* set. That is, sort the *active* set by $m$ as the major order in increasing order, and by $n$ as the minor order in decreasing order, then take out the first element and put it in the final ranked list.

Step 3: Repeat step 2 until the current *active* set is empty.

Step 4: If the final ranked list has less than 1000 documents, append more irrelevant documents drew from any individual list to the final ranked list.

The optimal ranking after reordering the sets is presented in table 18

| set | Optimal ranking |
|---|---|
| 1 | (0,1) $\{A_1\}$ |
| 2 | (1,3) $\{C_1,C_2,C_3,C_4\}$ |
| 3 | (1,1) $\{A_2,A_3\}$ |
| 4 | (2,1) $\{B_1,B_2,B_3\}$ |
| 5 | (1,0) $\{A_4\}$ |
| 6 | (1,0) $\{B_4\}$ |

Table 18: Optimal ranking.

The upper-bound average precision for the set of runs used for producing our official multilingual runs is 0.5177 with overall recall of 6392/8068. The performances of the direct merging and score-normalizing merging are far below the upper-bound performance that could possibly be achieved.

# 7 Conclusions

We have presented a technique for incorporating blind relevance feedback into a document ranking formula based on logistic regression analysis. The improvement in average precision brought by query expansion via blind relevance feedback ranges from 6.42% to 19.42% for monolingual retrieval runs, and from 10.85% to 29.36% for

cross-language retrieval runs. We have also presented a procedure to decompose German compounds and Dutch compounds. German decompounding improved the the average precision of German monolingual retrieval by 11.47%. Decompounding increased the average precision for cross-language retrieval to German from English or French. The increase ranges from 8.4% to 11.46%. For Dutch monolingual retrieval, decompounding increased the average precision by 4.10%, which is much lower than the improvement of 13.49% on CLEF 2001 test set. In summary, both blind relevance feedback and decompounding in German or Dutch have been shown to be effective in monolingual and cross-language retrieval. The amount of improvement of performance by decompounding varies from one set of topics to another. Three different translation resources, machine translators, parallel corpora, and bilingual dictionaries, were evaluated on cross-language retrieval. We found that the English-French CLIR performance of using parallel corpora was competitive with that of using commercial machine translators. Two different merging strategies in multilingual retrieval were evaluated. The simplest strategy of merging individual ranked lists of documents by unnormalized relevance score worked better than the one first normalizing the relevance score. To make the relevance scores of the documents from different collections as closely comparable as possible, we selected the same number of terms from the same number of top-ranked documents after the initial search for query expansion in all the runs that were combined to produce the unified ranked lists of documents in multiple languages. We used two machine translators to translate English topics to French, German, Italian and Spanish, and combined by topic the translations from the two translators. We reduced the term frequencies in the combined translated topics by half so that the combined translated topics are close in length to the source English topics. We presented an algorithm for generating the optimal ranked list of documents when the document relevances are known. The optimal performance can then be used to measure the performances of different merging strategies.

# 8 Acknowledgments

We would like to thank Vivien Petras for improving the German base dictionary. This research was supported by DARPA under research grant N66001-00-1-8911 (PI: Michael Buckland) as part of the DARPA Translingual Information Detection, Extraction, and Summarization Program (TIDES).

# References

[1] Y. Al-Onaizan et al. Statistical machine translation, final report, JHU workshop, 1999.

[2] P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19:263–312, June 1993.

[3] A. Chen. Multilingual information retrieval using english and chinese queries. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *Evaluation of Cross-Language Information Retrieval Systems: Second Workshop of the Cross-Language Evaluation Forum, CLEF-2001, Darmstadt, Germany, September 2001*, pages 44–58. Springer Computer Scinece Series LNCS 2406, 2002.

[4] W. S. Cooper, A. Chen, and F. C. Gey. Full text retrieval based on probabilistic equations with coefficients fitted by logistic regression. In D. K. Harman, editor, *The Second Text REtrieval Conference (TREC-2)*, pages 57–66, March 1994.

[5] M. Z. Daniel Karp, Yves Schabes and D. Egedi. A freely available wide coverage morphological analyzer for english. In *Proceedings of COLING*, 1992.

[6] A. Fox. *The Structure of German*. Clarendon Press, Oxford, 1990.

[7] W. A. Gale and K. W. Church. A program for aligning sentences in bilingual corpora. *Computational linguistics*, 19:75–102, March 1993.

[8] G. Grefenstette, editor. *Cross-language information retrieval*. Kluwer Academic Publishers, Boston, MA, 1998.

[9] D. Harman. Relevance feedback and other query modification techniques. In W. Frakes and R. Baeza-Yates, editors, *Information Retrieval: Data Structures & Algorithms*, pages 241–263. Prentice Hall, 1992.

[10] W. Lezius, R. Rapp, and M. Wettler. A freely available morphological analyzer, disambiguator and context sensitive lemmatizer for german. In *COLING-ACL'98*, pages 743–748, 1998.

[11] F. J. Och and H. Ney. A comparison of alignment models for statistical machine transl ation. In *COLING00*, pages 1086–1090, Saarbrücken, Germany, August 2000.

[12] C. Peters, editor. *Cross Language Information Retrieval and Evaluation: Proceedings of the CLEF 2000 Workshop*. Springer Computer Scinece Series LNCS 2069, 2001.

[13] C. Peters, editor. *Working Notes of the CLEF 2001 Workshop 3 September, Darmstadt, Germany*. September 2001.

[14] S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, pages 129–146, May–June 1976.

[15] E. Voorhees and D. Harman, editors. *The Seventh Text Retrieval Conference (TREC-7)*. NIST, 1998.

[16] E. Voorhees and D. Harman, editors. *The Eighth Text Retrieval Conference (TREC-8)*. NIST, 1999.